Contents

[Copy Files 2](#_Toc29889753)

[View and Edit the Model (Optional) 2](#_Toc29889754)

[Viewing and Editing the Model Using SAS ESP Studio (Optional) 2](#_Toc29889755)

[Upload Project 2](#_Toc29889756)

[Edit Filter 3](#_Toc29889757)

[Edit Output Schema of Join Window 4](#_Toc29889758)

[Edit Output Schema of Compute Window 5](#_Toc29889759)

[Edit Output Schema of Aggregate Window 6](#_Toc29889760)

[Download Project 7](#_Toc29889761)

[Viewing and Editing the Model with a Text Editor (Optional) 8](#_Toc29889762)

[Edit Filter 8](#_Toc29889763)

[Edit Output Schema of Join Window 8](#_Toc29889764)

[Edit Output Schema of Compute Window 9](#_Toc29889765)

[Edit Output Schema of Aggregate Window 10](#_Toc29889766)

[Executing the Model, Publishing the Files, and Subscribing to the Output 10](#_Toc29889767)

[Start the Model on the XML Server 10](#_Toc29889768)

[Publish Input Data 10](#_Toc29889769)

[Subscribe to the Output with SAS ESP Streamviewer 11](#_Toc29889770)

[Add a Chart 12](#_Toc29889771)

Instructions

These instructions assume you have SAS Event Stream Processing 6.2 installed.

# Copy Files

All required files are included in the SAS Event Stream Processing Examples directory ($DFESP\_HOME/examples). Create a server copy of the trades.xml, trades.csv, and traders.csv files, and a local copy of trades.xml for editing.

1. Copy the files from the SAS ESP Examples directory for this example ($DFESP\_HOME/examples/xml/trades\_xml) to a directory to which you have write access. Example:

cp $DFESP\_HOME/examples/xml/trades\_xml/\*.\* /home/sasdemo/trades

1. Download the trades.xml file to your local computer so it can be easily edited.

# View and Edit the Model (Optional)

Editing the model is optional. The model will run fine without any modifications.

You can view and edit your local copy of the trades.xml model in one of two ways:

* Upload the trades.xml file to SAS ESP Studio
* Open the trades.xml file with a text editor

## Viewing and Editing the Model Using SAS ESP Studio (Optional)

**NOTE: Ensure the SAS ESP Server is running.**

Use the following steps to view and edit the trades.xml model in SAS ESP Studio. You can skip this section and refer to the [Viewing and Editing the Model with a Text Editor](#_Viewing_and_Editing) section for instructions on editing the trades.xml model in a text editor.

### Upload Project

1. In the upper-right corner of SAS ESP Studio, click More actions to reveal the **More actions** menu and select **Upload projects**.  
   *The Upload Projects window appears.*
2. Click Upload project files.
3. Navigate to the file that contains the project that you want to upload and click **Open.**
4. Click **Upload**. The file is uploaded, and the **Upload Projects** window displays the file, project name, and a green checkmark if the upload was successful.
5. Click **Close** to close the **Upload Projects** window.

### Edit Filter

1. Double-click the project named **trades\_proj** to open it.
2. Click the **LargeTrades** window to select it and expand the **Filter** parameter on the right side.
3. Click Edit row to open the Rich-Expression Editor. You can use expressions, user-defined functions (global functions), and registered plug-in functions.

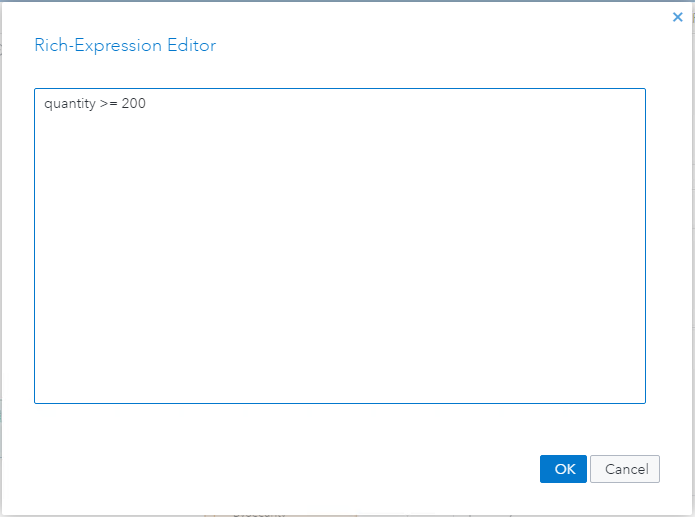


Figure 1 -- Connector Configuration Window

1. Edit the expression to be:

quantity >= 200

1. Click **OK**.

### Edit Output Schema of Join Window

1. Click the **AddTraderName** window to select it and expand the **Join Criteria** and **Join** **Conditions** parameters. Under Join Criteria, you will see the **Join type** is Left outer and under **Join Criteria** that traderID is joined with ID.
2. Click Output Schema to display the output schema and click Edit row to open the **Edit** **Output Schema – Non-Key Fields** window.

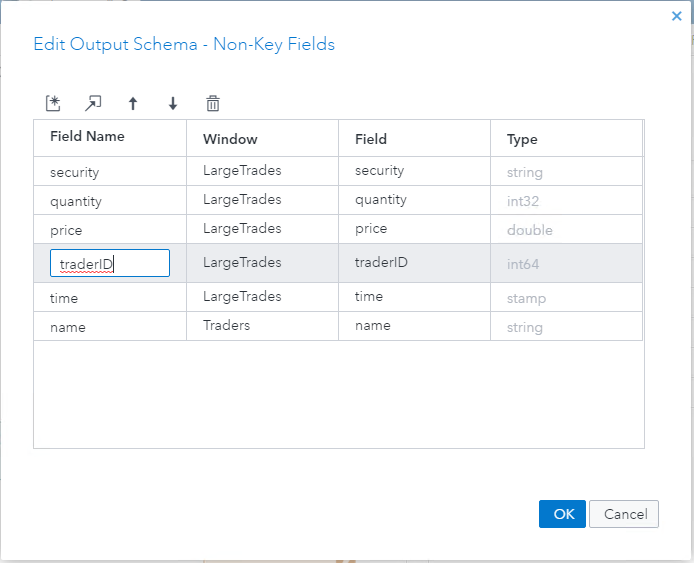


Figure 2 – Edit Output Schema – Non-Key Fields Window

1. Click the row containing traderID as the **Field Name** and then click  to remove the row from the stream.
2. Click **OK** to update the schema.

### Edit Output Schema of Compute Window

1. Click the **TotalCost** window to select it and, if necessary, click Output Schema to display the output schema. Click Edit row to open the **Output Schema** window in edit mode.

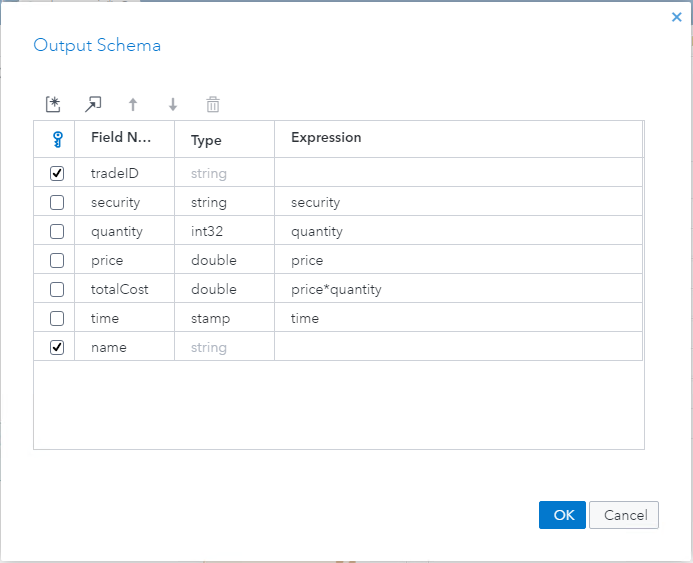
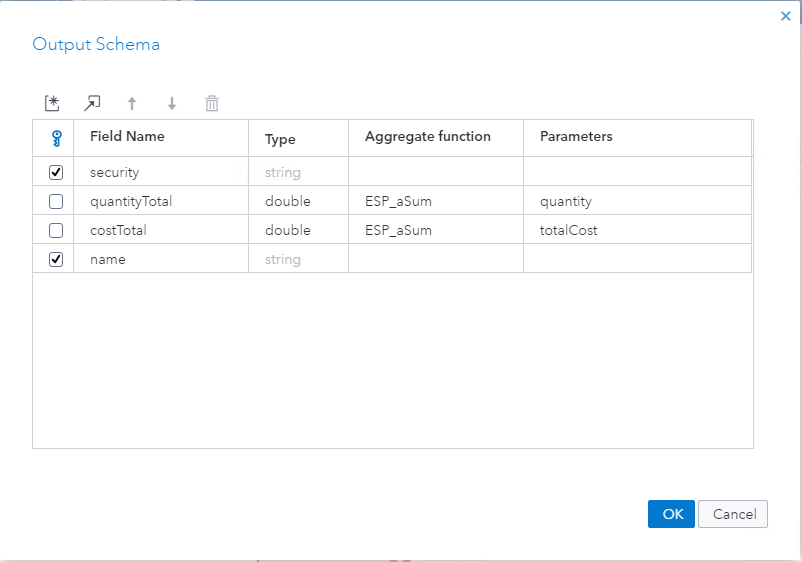


Figure 3 -- Output Schema Window

1. Click the row containing traderID as the **Field Name** and then click  to remove the row from the output.
2. Click the checkbox next to the row with the **Field Name** name to select it as an additional key field.
3. Click **OK** to update the schema.

### Edit Output Schema of Aggregate Window

1. Click the **BySecurity** window to select it and, if necessary, click Output Schema to display the output schema. Click Edit row to open the **Output Schema** window in edit mode.
2. Click Create new ESP server to add a new field.
3. Type **name** as the **Field Name** of the new field.
4. Click the checkbox next to the **name** field to select it as a key field. The Aggregate function and Parameters are set to missing by the system.



1. Click **OK** to update the schema.
2. Click Save to save your changes.

### Download Project

**NOTE: The trades model cannot be tested in SAS ESP Studio because there are no input data connectors to publish the data.**

1. Close the project.
2. Ensure **trades\_proj** is selected, click More actions to display the **More Actions** menu, and click **Download project**. A file named trades\_proj.xml is downloaded to your local Downloads folder.

## Viewing and Editing the Model with a Text Editor (Optional)

Use the following steps to view and edit the trades.xml model in a text editor. If you edited the model using SAS ESP Studio you can skip this section. Refer to the section, [Viewing and Editing the Model Using SAS ESP Studio](#_SAS_ESP_Studio), for instructions on editing the trades.xml model using SAS ESP Studio.

### Edit Filter

1. Locate the **window-filter** element.
2. Go to the **expression** element.
3. Edit the expression to be **quantity >= 200**.

<window-filter name='LargeTrades' index='pi\_EMPTY'>

<expression>quantity >= 200</expression>

</window-filter>

Figure 4 -- Window-Filter Element

### Edit Output Schema of Join Window

1. Locate the **window-join** element.
2. Go to the **output** element.
3. Delete the **field-selection** row for traderID.

<window-join name='AddTraderName' index='pi\_EMPTY'>

<join type='leftouter' no-regenerates='true'>

<conditions>

<fields left='traderID' right='ID'/>

</conditions>

</join>

<output>

<field-selection name='security' source='l\_security'/>

<field-selection name='quantity' source='l\_quantity'/>

<field-selection name='price' source='l\_price'/>

<field-selection name='traderID' source='l\_traderID'/>

<field-selection name='time' source='l\_time'/>

<field-selection name='name' source='r\_name'/>

</output>

</window-join>

Figure 5 -- Window-Join Element

### Edit Output Schema of Compute Window

1. Locate the **window-compute** element. Compute windows require all fields be in the schema element and all non-key fields in the output element.
2. Go to the **schema** element and delete the **field** element for traderID.
3. Go to the **output** element and delete the **field-expr** element for traderID.
4. Return to the **schema** element and add the attribute **key=’true’** to the **field** element for name.
5. Go to the **output** element and delete the **field-expr** element for name since it is now a key field.

<window-compute name='TotalCost' index='pi\_EMPTY'>

<description>

This is a compute window. The non-key fields are computed

in the output section.

</description>

<schema>

<fields>

<field name='tradeID' type='string' key='true'/>

<field name='security' type='string'/>

<field name='quantity' type='int32'/>

<field name='price' type='double'/>

<field name='totalCost' type='double'/>

<field name='traderID' type='int64'/>

<field name='time' type='stamp'/>

<field name='name' type='string' key='true'/>

</fields>

</schema>

<output>

<field-expr>security</field-expr>

<field-expr>quantity</field-expr>

<field-expr>price</field-expr>

<field-expr>price\*quantity</field-expr>

<field-expr>traderID</field-expr>

<field-expr>time</field-expr>

<field-expr>name</field-expr>

</output>

</window-compute>

Figure 6 -- Window-Compute Element

### Edit Output Schema of Aggregate Window

1. Locate the **window-aggregate** element.
2. Go to the **schema** element.
3. Add a **field** element for field name and indicate it is a key field.

<window-aggregate name='BySecurity'>

<schema>

<fields>

<field name='security' type='string' key='true'/>

<field name='name' type='string' key='true'/>

<field name='quantityTotal' type='double'/>

<field name='costTotal' type='double'/>

</fields>

</schema>

<output>

<field-expr>ESP\_aSum(quantity)</field-expr>

<field-expr>ESP\_aSum(totalCost)</field-expr>

</output>

</window-aggregate>

Figure 7 -- Window-Aggregate Element

1. Save your changes.
2. Upload the trades.xml file from your local computer to the server directory you are using.

# Executing the Model, Publishing the Files, and Subscribing to the Output

## Start the Model on the XML Server

Use the following command line syntax to start the trades.xml model on the XML server:

$DFESP\_HOME/bin/dfesp\_xml\_server -model file:///*yourpath*/trades.xml -http 61001 -pubsub 61002

## Publish Input Data

Use a file/socket adapter to publish the traders.csv file to the Traders window and the trades.csv file to the Trades window.

1. Open a new tab and type the following syntax to publish the traders.csv file to the Traders window using a file/socket adapter.

$DFESP\_HOME/bin/dfesp\_fs\_adapter -C type=pub,url=”DFesp://localhost:61002/trades\_proj/trades\_cq/Traders”,fstype=csv,fsname=/*yourpath*/traders.csv,blocksize=256

1. Open another new tab and type the following syntax to publish the trades.csv file to the Trades window using a file/socket adapter. The repeatcount=10000000 value-pair will cause the file to be published repeatedly allowing time to subscribe to the streaming output.

$DFESP\_HOME/bin/dfesp\_fs\_adapter -C type=pub,url=”DFesp://localhost:61002/trades\_proj/trades\_cq/Trades”,fstype=csv,fsname=/*yourpath*/trades.csv,blocksize=256,dateformat=%d/%b/%Y:%H:%M:%S,repeatcount=10000000

## Subscribe to the Output with SAS ESP Streamviewer

Use the following steps to subscribe to the **BySecurity** window using SAS ESP Streamviewer.

1. Start ESP Streamviewer using the following URL:

https://*Streamviewer-host-name*/SASEventStreamProcessingStreamviewer

1. On the ESP Streamviewer dashboard, click Show Model to open the **ESP Model Viewer**.

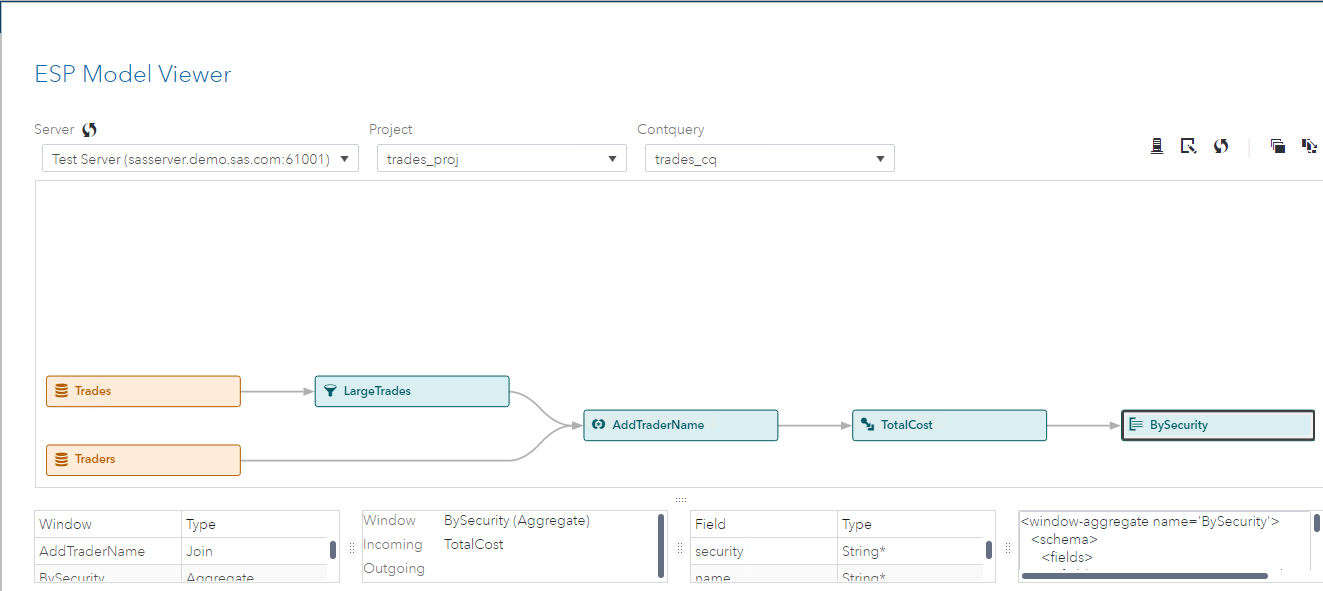


Figure 8 -- ESP Model Viewer

1. Click the **BySecurity** window to select it, and then click Add Updating Subscriber to add an updating subscriber to the dashboard. Click **Close** to close the **ESP Model Viewer**.  
   *The updating subscriber table for the BySecurity window appears on the dashboard.*

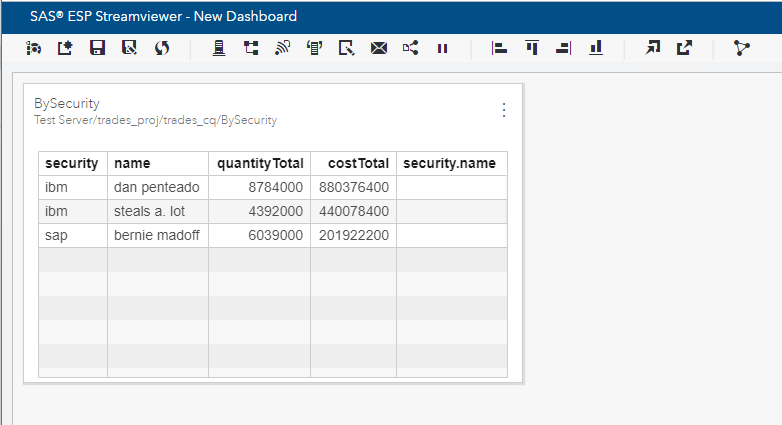


Figure 9 -- SAS ESP Streamviewer Dashboard Element

### Add a Chart

Use the following steps to add a line chart to the dashboard.

1. Click Window Menu at the top right of the table to display a menu of options and select **New** **Chart**.  
   *The New Chart window appears.*

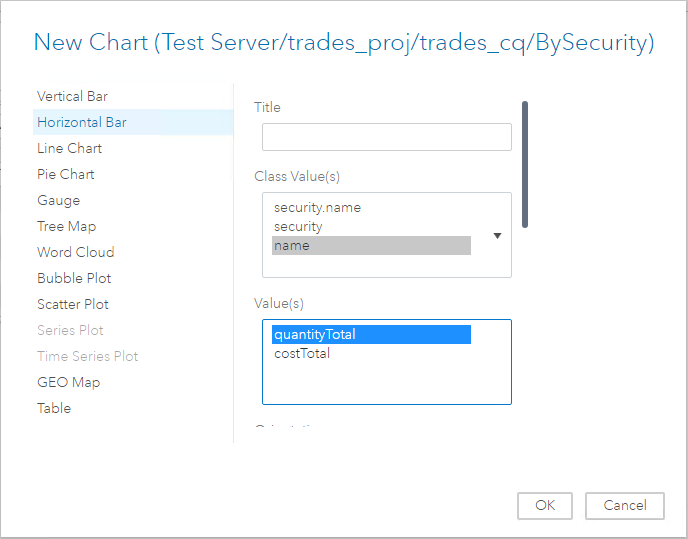


Figure 10 -- New Chart Window

1. From the left menu select **Horizontal Bar.**
2. Under **Class Value(s)** click **name** to select it.
3. Under **Value** select **quantityTotal**.
4. Click **OK** to close the **New Chart** window and display the horizontal bar chart on the dashboard.

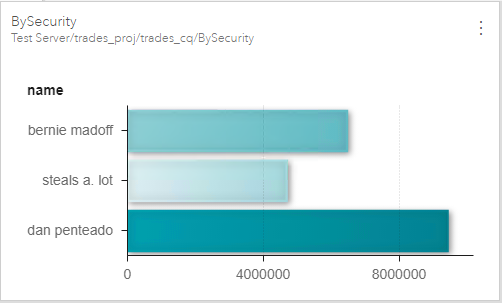


Figure 11 -- SAS ESP Streamviewer Dashboard Line Chart